Single machine batching problem to minimize the sum of completion times with number of batches and batch size limitations (Mathematical Optimization Theory and its Algorithm)

AUTHOR(S):

Muthusamy, Kanesan; Ishii, Hiroaki; Masuda, Teruo; Mohri, Shintaro

数理解析研究所講究録 1241 巻 2001 年 127-138

Single machine batching problem to minimize the sum of completion times with number of batches and batch size limitations

Kanesan Muthusamy [a, *] Hiroaki Ishii [a] Teruo Masuda [b] Shintaro Mohri [c]

[a] *Graduate School of Engineering, Osaka University, Osaka 565-0871. Japan.*
[b] *Faculty of Business Administration, Tezukayama University, Nara 631-8501. Japan.*
[c] *Faculty of Economics, Kobe Gakuin University, Kobe 651-2180. Japan.*

Abstract

We consider a single machine batching problem for identical jobs. Constant processing times and batch setup times are assumed together with number of batches and batch size limitations. We present a polynomially bounded algorithm that produces good near optimal solutions with respect to minimizing the sum of completion times.

Key-word: Scheduling, batching, polynomial algorithm, sum of completion times.

1 Introduction

This paper describes a single machine batch scheduling problem for identical jobs. A *batch* is defined as a set of jobs of the same type to be produced in a single setup. A constant *setup time* is required between consecutive batches. For a given batch, the number of jobs which are contained in the batch is called its *batch size.* Since processed items become available in batches, flow times are defined to be the same for all items in the same batch. The *number of batches* and *batch size* are assumed to be constraints. We present a polynomially bounded algorithm that determines near optimal number of batches and its batch sizes in which it produces good near optimal solutions with respect to minimizing the sum of completion times.

Over the past two decades, extensive research has been done on the batching problem [1, 2, 3, 4, 7, 8, 9, 10] but very little work has been done with number of batches and batch size limitations together. Examples include limited number of pallets and how many items a pallet can accommodate. Depending on the industry type, the number of pallets (or number of batches in this paper) can be fuzzy depending on pallet return or arrival from the end of operations. In this paper, we consider a fixed number of batches and batch size limitations.

This paper is organized as follows. In Section 2, we discuss some well-known work done on batching problems. In Section 3, we formulate the single machine batching problem and propose a polynomially bounded algorithm. In Section 4, we summarize the salient feature of this paper and discuss the directions for further research.

*Correspondence to: kanesan@ap.eng.osaka-u.ac.jp; Department of Applied Physics, Graduate School of Engineering, Osaka University, 2-1 Yamada-oka, Suita, Osaka 565-087. Japan. Fax: +81-6-6879-7871 .

2 Preliminaries

Previous works on a batching problem include Santos and Magazine [8], Dobson et al. [4], Naddef and Santos [7], Baker [1], Sung and Joo [10] and others [2, 3, 9]. We will review some of these works [4, 7, 10] done on a single machine batching problem.

Dobson et al. [4] proposed single product type baching problem on a single machine as follows:-

$$\min \quad \sum_{i=1}^{M}\sum_{k=1}^{i}(sq_i + tq_kq_i)$$

$$\text{s.t.} \quad \sum_{i=1}^{M} q_i = d,$$

$$q_i \geq 0, \quad i = 1, \ldots, M.$$

They obtained optimal number of batches and batch size as follows:

$$n = \left\lfloor \sqrt{\frac{1}{4} + (\frac{2dt}{s})} - \frac{1}{2} \right\rfloor \quad \text{and}$$

$$q_i^* = \begin{cases} \frac{d}{n} + \frac{s}{t}(\frac{n+1}{2}) - i\frac{s}{t} & \text{for} \quad i = 1, \ldots, n, \\ 0 & \text{for} \quad i = n+1, \ldots, M. \end{cases}$$

where

d - number of jobs,
s - setup time between batches,
t - job processing time,

Naddef and Santos [7] have also proposed a similar formulation for the single machine batching problem whose aim is to minimize the sum of completion times. They proposed a greedy algorithm called as one-pass batching algorithm. In fact, they considered the number of batches and batch size factors individually despite in real context these factors to be considered together. The optimum batching algorithm is modified to solve the problem of batching d jobs in exactly k batches ($d \geq k$). In the case of batch sizes upper bound β, an additional job can be added to a batch only if it contains less than β jobs. For more details, the reader is referred to [7].

Recently, Sung and Joo [10] considered the batch size restrictions for a single machine batching to minimize weighted mean flow time. The batch sizes are integers and restricted between lower and upper bounds. The lower bound is considered to incorporate the practical situation where each production operations requires a minimum amount of work load for keeping safety stock in the next work station. On the other hand, the upper bound is considered to incorporate any physical capacity restriction for production work on jobs. A feasibility condition and the optimal sequencing property is derived based on dynamic programming algorithm.

To our knowledge, so far, no research has been yet reported with constraints on number of batches and batch sizes simultaneously.

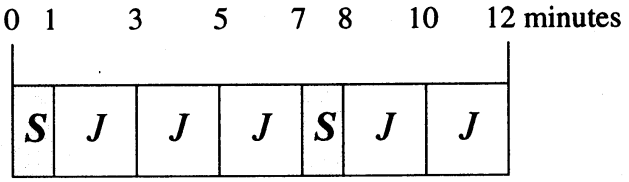
3 Problem formulation

We use the following notations:

p - job processing time,
s - setup time between consecutive batches,
d - number of jobs,
b_j - jth batch size, $j = 1, 2, \ldots, R$,
R - maximum allowable number of batches,
Q - maximum allowable batch size.

In our model, first, b_j is assumed to be a continuous variable despite of real fact that the parts are discrete, i.e. we treat the total number of parts d as homogeneous and divisible in any proportions, with R an upper bound on the total number of batches. Later, we use the continuous b_j's optimal solution and propose an approximation algorithm to obtain discrete solutions.

We describe a solution to the d-job batch by a vector B_d=$(b_1, \ldots, b_R)$ where b_j is the size of the jth batch (i.e. $b_j \neq 0, j = 1, \ldots, R$). We can describe a batching problem by the following example as shown in Fig.1. We take an identical part with five items to be processed. Each item requires two minutes to process and the setup time between batches is one minute. If we produce the five items into two batches, i.e. do three items in the first and two items in the second batch, the total sum of completion times for the batch is:



(3 items)(7 minutes) + (2 items)(12 minutes) = 45 items minutes.

Figure 1: Batching problem

The problem formulation of minimizing the sum of completion time on a single machine batch scheduling problem is as follows.

Let say, the initial batch with b_1 jobs is completed at $(s+b_1p)$, second batch at $(2s+(b_1+b_2)p)$. Similarly, each of the b_j jobs of the jth batch is completed at $(js + \sum_{l=1}^{j} b_l p)$. The final formulation to describe the value of our objective function is given as:

$$Z(B_d) = \sum_{j=1}^{R} b_j \left(js + \left[\sum_{l=1}^{j} b_l \right] p \right) \tag{3.1}$$

3.1 Continuous solution

Rewriting equation (3.1) in a slightly different form, we obtain the following problem Z :

Problem Z:

$$\min \quad Z = s\sum_{j=1}^{R} jb_j + \frac{1}{2}p\sum_{j=1}^{R} b_j^2 + \frac{1}{2}p\left(\sum_{j=1}^{R} b_j\right)^2, \tag{3.2}$$

$$\text{s.t.} \quad \sum_{j=1}^{R} b_j - d = 0, \qquad (\lambda), \tag{3.3}$$

$$b_j - Q \le 0, \quad j = 1, \ldots, R, \qquad (u_j), \tag{3.4}$$

$$-b_j \le 0, \quad j = 1, \ldots, R, \qquad (v_j), \tag{3.5}$$

The Kuhn-Tucker conditions for (3.2) - (3.5) are :-

$$\frac{\partial z}{\partial b_j} + \lambda\frac{\partial}{\partial b_j}\left(\sum_{j=1}^{R} b_j - d\right) + u_j\frac{\partial}{\partial b_j}(b_j - Q) + v_j\frac{\partial}{\partial b_j}(-b_j) = 0, \quad j = 1, \ldots, R,$$

$$sj + pb_j + pd + \lambda + u_j - v_j = 0, \quad j = 1, \ldots, R, \tag{3.6}$$

$$u_j(b_j - Q) = 0, \quad j = 1, \ldots, R, \tag{3.7}$$

$$v_j b_j = 0, \quad j = 1, \ldots, R, \tag{3.8}$$

$$u_j, v_j \ge 0, \quad j = 1, \ldots, R, \tag{3.9}$$

$$\sum_{j=1}^{R} b_j - d = 0, \tag{3.10}$$

$$b_j - Q \le 0, \quad j = 1, \ldots, R, \tag{3.11}$$

$$-b_j \le 0, \quad j = 1, \ldots, R, \tag{3.12}$$

where λ, u_j and v_j are the Lagrange multipliers associated with the three types of constraints. By the similar technique to [5, 6], solution of Kuhn-Tucker condition becomes as follows.

$$b_j(\lambda) = \max\left\{\min\left(\frac{-pd - \lambda - sj}{p}, Q\right), 0\right\} \quad (u_j = 0, v_j = 0) \tag{3.13}$$

$$u_j(\lambda) = \max\{-sj - pQ - pd - \lambda, 0\} \qquad (v_j = 0, b_j = Q) \tag{3.14}$$

$$v_j(\lambda) = \max\{sj + pd + \lambda, 0\} \quad (u_j = 0, b_j = 0) \tag{3.15}$$

$$b_j(\lambda) = \begin{cases} Q & \text{if} \quad \lambda \le -pd - sj - pQ, \\ \frac{-pd-\lambda-sj}{p} & \text{if} \quad -pd - sj - pQ < \lambda \le -pd - sj \\ 0 & \text{if} \quad \lambda > -pd - sj. \end{cases} \tag{3.16}$$

$$\sum_{j=1}^{R} b_j(\lambda) = d, \tag{3.17}$$

Now let $g(\lambda) = \sum_{j=1}^{R} b_j(\lambda)$. Then $g(\lambda)$ is a piece-wise linear and monotone non-increasing function of λ. Thus $\mathbf{b}$ can be found as $\mathbf{b} = \mathbf{b}(\lambda)$ from $\mathbf{b}(\lambda)$ satisfying $g(\lambda) = d$. Denoting breakpoints $-pd - sj - pQ$ and $-pd - sj$ with r_j and q_j , respectively and arranging them in the non-decreasing order, we let

$$y_1 < y_2 < \ldots < y_m \tag{3.18}$$

where m is the number of different r_j and q_j. Now, we are ready to propose Algorithm 1 for solving Problem Z.

Algorithm 1

Step 1: Set $l \leftarrow 1, r \leftarrow m, q \leftarrow \lfloor \frac{(l+r)}{2} \rfloor$ and $\lambda \leftarrow y_q$. Go to Step 2.

Step 2: Compute $g(\lambda) \leftarrow \sum_j \max\{\min(\frac{-pd-\lambda-sj}{p}, Q), 0\}$. If $g(\lambda) < d$, set $r \leftarrow q$ and go to Step 3; if $g(\lambda) = d$, then go to Step 5; if $g(\lambda) > d$, set $l \leftarrow q$ and go to Step 3.

Step 3: If $r - l = 1$, then go to Step 4; if $r - l \neq 1$, set $q \leftarrow \lfloor \frac{(l+r)}{2} \rfloor$ and $\lambda \leftarrow y_q$ and return to Step 2.

Step 4: Find λ such that $g(\lambda) = d$, set $\mathbf{b} \leftarrow \mathbf{b}(\lambda)$ and terminate.

Step 5: Set $\lambda \leftarrow y_q$ and $\mathbf{b} \leftarrow \mathbf{b}(\lambda)$, and terminate.

Upon obtaining the near optimal values of b_j, the batches are sequenced according to the increasing index of $j (j = 1, \ldots, R)$.

Theorem 1. Algorithm 1 finds near optimal solutions of $\mathbf{b}$ in at most $O(n\log n)$ computation times.

Proof : (Validity:) b_j's change their functional forms at $y_1, \ldots, y_m$ and Algorithm 1 checks all possible intervals for optimal condition $g(\lambda) = d$ by fully utilizing monotonicity of $g(\lambda)$. Therefore, termination conditions in Step 4 and Step 5 assure validity of Algorithm 1.
(Complexity:)

(i) Calculating $y_1, \ldots, y_m$ takes $O(n\log n)$ computational time since total number of r_j and q_j is $O(n)$ and sorting $O(n)$ elements takes at most $O(n\log n)$ computational time.

(ii) Step 1 takes at most $O(n)$ computational time.

(iii) Iteration number of Step 2 and Step 3 is at most $O(n\log n)$computational time.

(iv) Step 4 takes $O(n)$ computational time since solving a linear equation $g(\lambda) = d$ with respect to λ takes $O(n)$ computational time and setting $b \leftarrow b(\lambda)$ takes same order computational time.

(v) Step 5 takes $O(n)$ computational time.

(i)-(v) together prove complexity of Algorithm 1.

3.2 Approximation algorithm for discrete solutions

Let us define the followings:

$$b_j' = \lfloor b_j \rfloor, \qquad j = 1, \ldots, R, \tag{3.19}$$

We group the batches as follows:

Group $A = \{b_j' | b_j' = Q, \quad j = 1, \ldots, q\}$,
Group $B = \{b_j' | b_j' < Q, \quad j = q+1, \ldots, q+r\}$,

where

q = number of batches in Group A,
r = number of batches in Group B,

Then,

$$L = \sum_{j=q+1}^{R} (b_j - b_j') \tag{3.20}$$

The following simple Algorithm 2 will describe the procedures to obtain discrete solutions:

Algorithm 2

Add exactly one job per batch in Group B starting from it's first batch such that the new batch size, $b_j^* \leq Q, j = q+1, \ldots, q+r$. Once L number of jobs been added, terminate.

Let say, Z^{OPT} be the continuous solution, Z' be the solution obtained by the equation (3.19) and Z^* be the discrete solution. Algorithm 2 will produce discrete solutions, which are bounded by ΔZ as follows:-

$$\Delta Z = Z^* - Z^{OPT} \tag{3.21}$$

$$\begin{aligned}
\Delta Z \;=\; & s\sum_{j=1}^{q} j b_j' + \frac{1}{2}p\sum_{j=1}^{q} b_j'^2 + \frac{1}{2}p\left(\sum_{j=1}^{q} b_j'\right)^2 \\
& + s\sum_{j=q+1}^{q+L} j(b_j'+1) + \frac{1}{2}p\sum_{j=q+1}^{q+L} (b_j'+1)^2 + \frac{1}{2}p\left(\sum_{j=q+1}^{q+L} b_j' + 1\right)^2 \\
& + s\sum_{j=q+L+1}^{R} j b_j' + \frac{1}{2}p\sum_{j=q+L+1}^{R} b_j'^2 + \frac{1}{2}p\left(\sum_{j=q+L+1}^{R} b_j'\right)^2 \\
& - \left[s\sum_{j=1}^{R} j b_j + \frac{1}{2}p\sum_{j=1}^{R} b_j^2 + \frac{1}{2}p\left(\sum_{j=1}^{R} b_j\right)^2\right]
\end{aligned}$$

$$\Delta Z = s\sum_{j=q+1}^{q+L} j(b_j' - b_j) + \frac{1}{2}p\sum_{j=q+1}^{q+L}(b_j'^2 - b_j^2) + \frac{1}{2}p\left[\left(\sum_{j=q+1}^{q+L} b_j'\right)^2 - \left(\sum_{j=q+1}^{q+L} b_j\right.\right.$$
$$+ s\sum_{j=q+1}^{q+L} j + p(1+L)\sum_{j=q+1}^{q+L} b_j' + \frac{1}{2}pL(1+L) + s\sum_{j=q+L+1}^{R} j(b_j' - b_j)$$
$$+ \frac{1}{2}p\sum_{j=q+L+1}^{R}(b_j'^2 - b_j^2) + \frac{1}{2}p\left[\left(\sum_{j=q+L+1}^{R} b_j'\right)^2 - \left(\sum_{j=q+L+1}^{R} b_j\right)^2\right]$$

Let consider the worst-case where the difference between b_j' and b_j are approximately closer to one. Then,

$$\Delta Z = 2s\sum_{j=q+1}^{q+L} j + 2s\sum_{j=q+L+1}^{R} j + \frac{1}{2}pL(1+L) + \frac{1}{2}p\sum_{j=q+1}^{q+L}(b_j' + b_j)$$
$$+ \frac{1}{2}p\left[\left(\sum_{j=q+1}^{q+L} b_j'\right)^2 - \left(\sum_{j=q+1}^{q+L} b_j\right)^2\right] + p(1+L)\sum_{j=q+1}^{q+L} b_j'$$
$$+ \frac{1}{2}p\sum_{j=q+L+1}^{R}(b_j' + b_j) + \frac{1}{2}p\left[\left(\sum_{j=q+L+1}^{R} b_j'\right)^2 - \left(\sum_{j=q+L+1}^{R} b_j\right)^2\right]$$

$$\Delta Z = \frac{1}{2}s\left(R^2 + L^2 + 2QL + R + L - q^2 - q\right) + \frac{1}{2}pL(1+L)$$
$$+ \frac{1}{2}p\sum_{j=q+1}^{R} b_j' + \frac{1}{2}p\sum_{j=q+1}^{R} b_j + \frac{1}{2}p\left[\left(\sum_{j=q+1}^{q+L} b_j'\right)^2 - \left(\sum_{j=q+1}^{q+L} b_j\right)^2\right] \quad (3.22)$$
$$+ p(1+L)\sum_{j=q+1}^{q+L} b_j' + \frac{1}{2}p\left[\left(\sum_{j=q+L+1}^{R} b_j'\right)^2 - \left(\sum_{j=q+L+1}^{R} b_j\right)^2\right]$$

From (3.20),

$$\sum_{q+1}^{R} b_j' = d - L - qQ \quad (3.23)$$

Substitute (3.23) and (3.3) into (3.22), we get the following:

$$\Delta Z = \frac{1}{2}s\left(R^2 + L^2 + 2QL + R + L - q^2 - q\right) + \frac{1}{2}pL(1+L) \quad (3.24)$$
$$+ p(d - L - qQ) + \frac{1}{2}pL + \frac{1}{2}p\left[\sum_{q+1}^{q+L} b_j'\left(\sum_{q+1}^{q+L)} b_j' + \frac{1}{2} + \frac{L}{2}\right)\right]$$
$$+ \frac{1}{2}p\left(\sum_{q+L+1}^{R} b_j'\right) - \frac{1}{2}p\left((d - qQ)^2 - 2\sum_{q+1}^{q+L} b_j\sum_{q+L+1}^{R} b_j\right)$$

From (3.16), we obtain the followings:

$$Qq + \sum_{j=q+1}^{q+r} \left(\frac{-pd - \lambda - sj}{p} \right) = d \tag{3.25}$$

Solving the above equation, we get,

$$\lambda = \frac{q}{r}(pQ - rs) - \frac{1}{r}(r+1)(pd + \frac{1}{2}rs) \leq -pd - sj - pQ \tag{3.26}$$

Assuming $q = j$, then,

$$j \leq r\left(\frac{s}{2pQ}(r+1) - 1\right) + \frac{d}{Q} \tag{3.27}$$

and

$$q = \left\lfloor r\left(\frac{s}{2pQ}(r+1) - 1\right) + \frac{d}{Q} \right\rfloor \tag{3.28}$$

From (3.16),

$$-pd - sj - pQ < \lambda < -pd - sj$$
$$\frac{1}{s}(-\lambda - pd) - \frac{pQ}{s} < j < \frac{1}{s}(-\lambda - pd)$$

Then, the value of r is given by the following equation:

$$r = \begin{cases} \lfloor \frac{pQ}{s} \rfloor - 1 \\ \lfloor \frac{pQ}{s} \rfloor \end{cases}$$

The maximum number of batches in group B is given as follows:

$$r = \left\lfloor \frac{pQ}{s} \right\rfloor \tag{3.30}$$

Substitute (3.30) into (3.28), we obtain the final form of q as follows:

$$q = \left\lfloor \left(\lfloor \frac{pQ}{s} \rfloor \right) \left[\frac{s}{2pQ} \left(\lfloor \frac{pQ}{s} \rfloor + 1 \right) - 1 \right] + \frac{d}{Q} \right\rfloor \tag{3.31}$$

From (3.30), the maximum of L is given by:

$$L \leq \left\lfloor \frac{pQ}{s} \right\rfloor - 1 \tag{3.32}$$

From (3.20), we obtain the followings:

$$\begin{aligned}
\sum_{j=1}^{R} b_j - \sum_{j=1}^{R} b'_j &= L \\
d - \sum_{j=1}^{R} b'_j &= L \\
\sum_{j=1}^{R} b'_j &= d - L \\
\sum_{j=1}^{q} b'_j + \sum_{j=q+1}^{q+L} b'_j + \sum_{j=q+L+1}^{R} b'_j &= d - L
\end{aligned}$$

$$\sum_{j=q+1}^{q+L} b'_j = d - L - qQ - \sum_{j=q+L+1}^{R} b'_j \tag{3.33}$$

By the similar way, we obtain the following from (3.3):

$$\sum_{j=q+1}^{q+L} b_j = d - qQ - \sum_{j=q+L+1}^{R} b_j \tag{3.34}$$

There are two cases for (3.33) and (3.34) as follows:

Case I:

$$\sum_{j=q+1}^{q+L} b'_j > \sum_{j=q+L+1}^{R} b'_j \tag{3.35}$$

$$\sum_{j=q+1}^{q+L} b_j > \sum_{j=q+L+1}^{R} b_j \tag{3.36}$$

$$\sum_{j=q+L+1}^{R} b'_j = 0 \tag{3.37}$$

$$\sum_{j=q+L+1}^{R} b'_j \geq 1 \tag{3.38}$$

$$\sum_{j=q+L+1}^{R} b_j \geq 0 \tag{3.39}$$

When we consider conditions (3.37) and (3.39), the worst-case upper bound is given as follows:

$$\begin{aligned}\Delta Z^{UB} \leq\ & \frac{1}{2}s\left(R^2+L^2+2qL+R+L-q^2-q\right)+\frac{1}{2}pL\left(1+L\right)+p(d-L-qQ)\\ & +\frac{1}{2}pL+\frac{1}{2}p\left(d-L-qQ\right)\left[(d-L-qQ)+\frac{1}{2}+\frac{L}{2}\right]\\ & -\frac{1}{2}p\left(d-qQ\right)^2\end{aligned} \tag{3.40}$$

where, $L \leq \left\lfloor \frac{pQ}{s} \right\rfloor - 1$ and

$q = \left\lfloor \left(\lfloor \frac{pQ}{s} \rfloor\right)\left[\frac{s}{2pQ}\left(\lfloor \frac{pQ}{s} \rfloor + 1\right) - 1\right] + \frac{d}{Q} \right\rfloor$

When we consider conditions (3.38) and (3.39), the worst-case upper bound is given as follows:

$$\begin{aligned}\Delta Z^{UB} \leq\ & \frac{1}{2}s\left(R^2+L^2+2qL+R+L-q^2-q\right)+\frac{1}{2}pL\left(1+L\right)+p(d-L-qQ)\\ & +\frac{1}{2}pL+\frac{1}{2}p\left(d-L-qQ-1\right)\left[(d-L-qQ-1)+\frac{1}{2}+\frac{L}{2}\right]\\ & +\frac{1}{2}p-\frac{1}{2}p\left(d-qQ\right)^2\end{aligned} \tag{3.41}$$

Case II:

$$\sum_{j=q+1}^{q+L} b_j' \leq \sum_{j=q+L+1}^{R} b_j' \tag{3.42}$$

$$\sum_{j=q+1}^{q+L} b_j \leq \sum_{j=q+L+1}^{R} b_j \tag{3.43}$$

Using (3.42) and (3.43), the worst-case upper bound is obtained as follows:

$$\begin{aligned}\Delta Z^{UB} \leq\ & \frac{1}{2}s\left(R^2+L^2+2qL+R+L-q^2-q\right)+\frac{1}{2}pL\left(1+L\right)\\ & +p(d-L-qQ)+\frac{1}{2}pL+\frac{1}{8}p\left(d-L-qQ\right)\left(d-qQ+1\right)\\ & +\frac{1}{8}p\left(d-L-qQ\right)^2-\frac{1}{4}p\left(d-qQ\right)^2\end{aligned} \tag{3.44}$$

where, $L \leq \left\lfloor \frac{pQ}{s} \right\rfloor - 1$ and

$q = \left\lfloor \left(\lfloor \frac{pQ}{s} \rfloor\right)\left[\frac{s}{2pQ}\left(\lfloor \frac{pQ}{s} \rfloor + 1\right) - 1\right] + \frac{d}{Q} \right\rfloor$

3.3 Examples

To illustrate the new heuristic, we use four examples as shown in Appendix I. Discrete solutions are obtained in Example 1 by using Algorithm 1. The results on Examples two, three and four indicate that the sum of completion times difference between the discrete and continuous solutions are at very minimal level. The worst-case upper bound obtained are quite loose compared to the actual difference. This was due to the assumptions made especially in (3.22) where the difference between b_j' and b_j are approximately closer to one. In overall, our algorithm may provide good discrete solutions.

4 Conclusion

This paper has introduced single machine batch scheduling problem that consider the number of batches and batch size limitations. We proposed a polynomially bounded algorithm that determines near optimal number of batches and its batch sizes in which it produces good near optimal solutions with respect to minimizing the sum of completion times. The investigation was motivated by these two factors which are very common in many real situations, manufacturing facilities in particular. We also proposed continuous and discrete solutions to the original problem. The proposed worst-case upper bound is quite loose and that can obviously be improved. We have studied the single machine and identical products problems but in real context it may be typically multi-machine and multi-product problem with many complicating factors, eg. number of batches constraint that can be fuzzy. Further research will address these issues.

References

[1] K.R.Baker, Sceduling The Production Of Components At A Common Facility, IIE Transactions 20(1) (1988) 32-35.

[2] E.G.Coffman, Jr., A.Nozari and M.Yannakakis, Optimal scheduling of products with two subassemblies on a single machine, Operations Research 37(3) (1989) 426-436.

[3] E.G.Coffman, Jr., M.Yannakakis, M.J.Magazine and C.Santos, Batch sizing and job sequencing on a single machine, Annals of Operations Research 26 (1990) 135-147.

[4] G.Dobson, U.S.Karmarkar and J.L.Rummel, Batching to minimize flow times on one machine, Management Science33(6) (1987) 784-799.

[5] R.Helgason, J.Kennington and H.Lall, A polynomially bounded algorithm for a singly constrained quadratic program, Mathematical Programming 16 (1980) 338-343.

[6] H.Ishii and T.Nishida, Stochastic Linear Knapsack Problem, Technology Reports of The Osaka University 32(1633) (1982) 25-30.

[7] D.Naddef and C.Santos, One-pass batching algorithms for the one-machine problem, Discrete Applied Mathematics 21 (1988) 133-145.

[8] C.Santos and M.Magazine, Batching in single operation manufacturing systems, Operations Research Letters 4(3) (1985) 338-343.

[9] D.F.Shallcross, A polynomial algorithm for a one machine batching problem, Operations Research Letters 11 (1992) 213-218.

[10] C.S.Sung and U.G.Joo, Batching to minimize weighted mean flow time on a single machine with batch size restrictions, Computers and Industrial Engineering 32(2) (1997) 333-340.

Appendix I

	Example 1	Example 2	Example 3	Example 4
d	10	15	20	30
p	1	2	2	3
s	2	3	4	5
Q	3	5	5	8
R	4	5	5	6
b_1	3	5	5	8
b_2	3	4.75	5	7.73
b_3	3	3.25	5	6.07
b_4	1	1.75	3.5	4.4
b_5	n	0.25	1.5	2.73
b_6	n	n	n	1.07
Z^{OPT}	108	383.75	695.5	2030.09
$b_1^{'}$	n	5	5	8
$b_2^{'}$	n	4	5	7
$b_3^{'}$	n	3	5	6
$b_4^{'}$	n	1	3	4
$b_5^{'}$	n	0	1	2
$b_6^{'}$	n	n	n	1
b_1^*	n	5	5	8
b_2^*	n	5	5	8
b_3^*	n	4	5	7
b_4^*	n	1	4	4
b_5^*	n	0	1	2
b_6^*	n	n	n	1
Z^*	n	385	696	2032
ΔZ	n	1.25	0.5	1.91
r	1	3	2	4
L	0	2	1	3
q	3	1	3	1
ΔZ^{UB}	n	41.5	51	40

n - *not applicable*

Table 1: Examples